IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1-12 and ADD new claims 13-19 in accordance with the following:

1. (**currently amended**) A low density parity check (LDPC) error correction method comprising:

generating an m*1 resultant matrix ~~(m*1)~~ by performing an XOR operation and a modular 2 operation with respect to an m*n LDPC matrix ~~(m*n)~~ and an n*1 code word vector ~~(n*1)~~;

determining whether a decoding of the code word vector succeeded based on ~~the basis of~~ the resultant matrix;

if ~~it is~~ determined that the decoding failed, detecting a code word bit, in which an error is ~~generated, in~~generated in the code word vector ~~on the basis of~~based on correlations of components of the LDPC matrix, the code word vector, and the resultant matrix; and

correcting the error by changing a binary value of the detected code word bit.

2. (**currently amended**) The method of claim 1, wherein the detecting of the code word bit in which the error is generated comprises:

checking whether ~~the same~~a column vector having same bit values as the resultant matrix exists in the LDPC matrix; and

if the ~~same~~ column vector having the same bit values as the resultant matrix exists in the LDPC matrix, detecting a code word bit of the code word vector corresponding to a column number of the ~~searched~~ column vector of the LDPC matrix having the same bit values as the resultant matrix ~~in the code word vector as~~ to be ~~an error generation~~ a code word bit in which an error is generated.

3. (**currently amended**) The method of ~~claim 1~~claim 2, wherein the detecting of the code word bit in which the error is generated further comprises:

if the ~~same~~ column vector having the same bit values as the resultant matrix does not

exist in the LDPC matrix, determining that the decoding failed and ending without correcting the generated error ~~correction process~~.

4. (**currently amended**) The method of claim 1, further comprising:

detecting ~~the~~ a number of generated 1-bit errors based on the number of 1s included in the resultant matrix.

5. (**currently amended**) The method of claim 4, wherein the detecting the number of generated 1-bit errors comprises:

checking whether the number of 1s included in the resultant matrix is the same as a column weight of the LDPC matrix; and

if the number of 1s included in the resultant matrix is not the same as a column weight of the LDPC matrix, ~~ending the error correction process by~~ determining that the decoding failed and ending without correcting the error.

6. (**currently amended**) The method of ~~claim 3~~claim 2, further comprising:

generating a new m*1 resultant matrix ~~(m*1)~~ by performing the XOR operation and modular 2 operation with respect to the m*n LDPC matrix ~~(m*n)~~ and a corrected n*1 code word vector ~~(n*1)~~ in which ~~the~~ a binary value of the detected code word bit is changed; and

determining whether ~~a~~ decoding of the corrected code word vector succeeded based on the ~~basis of the~~ new m*1 resultant matrix.

7. (**currently amended**) ~~An error determination~~A method of determining an error, comprising:

~~a first decoding success/failure determination step of~~ generating an m*1 resultant matrix ~~(m*1)~~ by multiplying an m*n LDPC matrix ~~(m*n)~~ by an n*1 code word vector ~~(n*1)~~ and determining whether a decoding of the code word vector succeeded based on the ~~basis of the~~ resultant matrix;

and ~~a second decoding success/failure determination step of,~~ if ~~it is~~ determined that the decoding failed, determining again whether the decoding succeeded based on ~~the basis of the~~ a number of 1s included in the resultant matrix.

8. (**currently amended**) The method of claim 7, ~~wherein the second decoding of success/failure determination step comprises~~further comprising:

determining that the decoding failed if the number of 1s included in the resultant matrix is not ~~the same as~~equal to a column weight of the LDPC matrix.

9. (**currently amended**) The method of claim 7, further comprising:

~~a third decoding success/failure determination step of~~ determining whether the decoding succeeded or failed by checking whether ~~the~~ a same column matrix as the resultant matrix exists in the LDPC matrix.

10. (**currently amended**) A low density parity check (LDPC) error correction apparatus comprising:

a decoding success/failure checking unit generating an m*1 resultant matrix ~~(m*1)~~ by performing an XOR operation and a modular 2 operation with respect to an m*n LDPC matrix ~~(m*n)~~ and an n*1 code word vector ~~(n*1)~~ and determining whether a decoding of the code word vector succeeded based on the ~~basis of the~~ resultant matrix;

an error location detector searching the LDPC matrix for a same column vector as the resultant matrix ~~in the LDPC matrix~~ and, if the same column vector exists in the LDPC matrix, detecting an error location in the code word vector by detecting a column number of the same column vector; and

a binary value changing unit correcting the error by changing a binary value of a code word bit in the code word vector corresponding to the detected column number ~~in the code word vector~~.

11. (**currently amended**) The apparatus of claim 10, wherein: ~~the error location detector,~~

if the same column ~~matrix~~ vector as the resultant matrix does not exist in the LDPC matrix, the error location detector determines that the decoding failed ~~and ends the error correction process~~.

12. (**currently amended**) The apparatus of claim 10, further comprising:

an error count detector determining whether ~~the~~ a number of 1s included in the resultant

matrix is the same as a column weight of the LDPC matrix and, if ~~they~~ the number of 1s is not equal to the column weight ~~are not the same~~, ~~ending the error correction process by~~ determining that the decoding failed.

13. (**new**) A method of correcting a 1 bit error in a code word vector using a low density parity check (LDPC) matrix, the method comprising:

generating a resultant matrix by performing an XOR operation and a modular 2 operation with respect to the LDPC matrix and the code word vector;

identifying a number of a column of the LDPC matrix having bit values corresponding to bit values of the resultant matrix, respectively; and

correcting the codeword by changing a binary value of the bit number of the codeword corresponding to the identified column number of the LDPC matrix.

14. (**new**) The method of claim 13, further comprising:

determining that the error is not the 1 bit error if the bit values of the resultant matrix do not correspond to the bit values of any column of the LDPC matrix.

15. (**new**) The method of claim 13, further comprising:

ending the method without correcting the error if the bit values of the resultant matrix do not correspond to the bit values of any column of the LDPC matrix

16. (**new**) A method of correcting a 1 bit error in a code word vector using a low density parity check (LDPC) matrix, the method comprising:

generating a resultant matrix by performing an XOR operation and a modular 2 operation with respect to the LDPC matrix and the code word vector;

determining whether a number of 1s in the resultant matrix is equal to a column weight of the LDPC matrix;

identifying a number of a column of the LDPC matrix having bit values corresponding to bit values of the resultant matrix, respectively, if the number of 1s in the resultant matrix is equal to the column weight of the LDPC matrix; and

correcting the codeword by changing a binary value of the bit number of the code word vector corresponding to the identified column number of the LDPC matrix.

17. (**new**) The method of claim 16, further comprising:

ending the method without correcting the error if the number of 1s in the resultant matrix is not equal to the column weight of the LDPC matrix.

18. (**new**) The method of claim 16, further comprising:

ending the method without correcting the error if no column of the LDPC matrix has bit values corresponding to the bit values of the resultant matrix.

19. (**new**) The method of claim 16, further comprising:

generating a second resultant matrix by performing another XOR operation and another modular 2 operation with respect to the LDPC matrix and the corrected code word vector; and

determining that the error has been corrected if all bits of the second resultant matrix have a zero value.